

## CISC 106 Summer 2014 Project – Part I

**Submission Deadline: 07/30/2014 11:55pm**

- \* **Form teams of two people. If you are not willing to work in a group, you can work by yourself.**
- \* **Write at least 3 test cases for each function you implement. If the function gets user input, you do not need to write test cases.**
- \* **Write both team members name at the top of project files.**

In this project you are going to implement a version of the battleship game. If you do not know the rules of the game, read the following Wikipedia page about the game ([http://en.wikipedia.org/wiki/Battleship\\_\(game\)](http://en.wikipedia.org/wiki/Battleship_(game))). You can also play the online version of the battleship game at <http://www.battleshiponline.org/>.

The game that you will develop is going to be a little more complicated than the regular battleship game, but the main logic is going to be the same. So, before continuing to read this document, first make sure you know how to play the battleship game and spend some time thinking about how you would design the classic battleship game. Let me give you a hint. When I played the battleship at <http://www.battleshiponline.org/> the program first showed me a grid board which has two pieces; one piece for me and one piece for my opponent (a computer) and asked me to place my ships to the board. Then the game continued with firing each other's board on a turn-by-turn basis. When there was no ship left on my opponent's board, the computer informed me that I won. So what I noticed is that the game has 3 stages which are: creating and initializing the boards, inserting the ships, and playing the game in turns. You can assume that the game you are going to develop will have the same stages.

You should first think about how you are going to represent the two boards (one for first user and one for second user) in your program and how you are going to simulate the turn shifts before embarking on coding. A simple solution to representing boards and turn shifts can be using two dimensional nested lists (for boards) and loops (for turn shifts).

### CISC 106 Battleship Game Requirements



